

Discrete-event simulation of fluid stochastic Petri nets*

Gianfranco Ciardo

Dept. of Computer Science
College of William and Mary
Williamsburg, VA 23187-8795
`ciardo@cs.wm.edu`

David Nicol†

Dept. of Computer Science
Dartmouth College
Hanover, NH 03755
`nicol@cs.dartmouth.edu`

Kishor S. Trivedi‡

Center for Advanced Computing and Communication
Dept. of Electrical and Computer Engineering
Duke University
Durham, NC 27708-0291
`kst@egr.duke.edu`

Abstract

The purpose of this paper is to describe a method for simulation of recently introduced fluid stochastic Petri nets. Since such nets result in rather complex set of partial differential equations, numerical solution becomes a formidable task. Because of a mixed, discrete and continuous state space, simulative solution also poses some interesting challenges, which are addressed in the paper.

*This research was supported in part by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the authors were in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001.

†The work of the second author was supported in part by the National Science Foundation under grants CCR-9201195 and NCR-9527163.

‡The work of the third author was supported in part by the National Science Foundation under grant NSF-EEC-94-18765.

1 Introduction

Stochastic Petri nets provide a convenient and concise method of describing discrete-event dynamic systems [1, 4, 6, 12, 15]. One of the difficulties encountered while using stochastic Petri nets is that the underlying reachability graph tends to be very large in practical problems. Drawing a parallel with fluid flow approximations in performance analysis of queueing systems [3, 7, 11], SPNs have been extended to include fluid (or continuous) places where fluid can be used to approximate a large number of discrete tokens. Armed with such Fluid Stochastic Petri nets (FSPNs), we can also model physical systems that contain continuous fluid-like quantities which are controlled by discrete logic.

Fluid stochastic Petri nets were introduced by Trivedi and Kulkarni in [14]. The original model was considerably enhanced in [9]. The purpose of this paper is twofold: first we further extend the formalism in [9] to make it more useful and, second, we explore the use of simulation as a solution method for FSPNs. The extensions to the FSPN formalism we propose include:

- Fluid impulses associated with both immediate and timed transition firings.
- Guards on immediate transitions, dependent on fluid levels, not just on the discrete marking.

In the process of using simulation as a solution method, several innovations are needed. In this direction, the contributions of this paper include:

- Generation of random deviates based on a non-homogeneous Poisson process.
- Interleaving of ODE solution for fluid places with simulation of discrete events in the FSPN.
- Definition of restrictions under which one can integrate the change of fluid levels using built-in closed-form results, such as decoupled behavior and special classes of functions for the fluid rates.

After introducing the FSPN model in the next section, we describe the method of simulation for the most general case in Section 3. The simplest case of uncoupled behavior of different fluid places is taken up in Section 4 while two other cases are discussed in Sections 5 and 6, respectively. Examples are provided in Section 7.

2 Fluid stochastic Petri nets

Sets are denoted by upper case calligraphic letters. In particular, \mathcal{N} , \mathcal{R} , and \mathcal{R}_0 indicate the natural, real, and nonnegative numbers, respectively.

For simplicity, we only address exponentially distributed firing times. Generally distributed firing times are clearly useful, and, in connection with discrete-event simulation, might not add much complexity to the solution. However, the interruption policies (what happens to the remaining firing time of transitions when one of them fires) can require very complex definitions in full generality [5, 13]. This is not the case with the exponential distribution, due to its memoryless property. If we assume that the firing rate of each transition, if marking-dependent, is reevaluated in each marking where it is enabled, the time elapsed since the transition first became enabled does not affect the future evolution of the FSPN.

A fluid stochastic Petri net (FSPN) is a tuple $(\mathcal{P}^D, \mathcal{P}^C, \mathcal{T}^T, \mathcal{T}^I, a, f, g, \lambda, w, b, \mathbf{m}^0, \mathbf{x}^0)$, where:

- $\mathcal{P}^D = \{p_1, \dots, p_{|\mathcal{P}^D|}\}$ and $\mathcal{P}^C = \{q_1, \dots, q_{|\mathcal{P}^C|}\}$ are two disjoint and finite sets of places. Let $\mathcal{P} = \mathcal{P}^D \cup \mathcal{P}^C$. A (discrete) place $p \in \mathcal{P}^D$ is drawn with a single circle and can contain a number of tokens $\mathbf{m}_p \in \mathcal{N}$. A (continuous) place $q \in \mathcal{P}^C$ is drawn with two concentric circles and can contain a level of fluid $\mathbf{x}_q \in \mathcal{R}_0$. The marking, or state, of the FSPN is given by a pair of vectors describing the contents of each place, $(\mathbf{m}, \mathbf{x}) \in \hat{\mathcal{S}} = \mathcal{N}^{|\mathcal{P}^D|} \times \mathcal{R}_0^{|\mathcal{P}^C|}$. We call $\hat{\mathcal{S}}$ the “potential state space”, as opposed to the “actual state space” $\mathcal{S} \subseteq \hat{\mathcal{S}}$, the set of markings actually reachable during the evolution of the FSPN. The marking (\mathbf{m}, \mathbf{x}) evolves in time, which we indicate by τ , so, formally, we can think of it as a stochastic process $\{(\mathbf{m}(\tau), \mathbf{x}(\tau)), \tau \geq 0\}$.
- $\mathcal{T}^T = \{t_1, \dots, t_{|\mathcal{T}^T|}\}$ and $\mathcal{T}^I = \{u_1, \dots, u_{|\mathcal{T}^I|}\}$ are two disjoint and finite sets of transitions. Let $\mathcal{T} = \mathcal{T}^T \cup \mathcal{T}^I$. A (timed) transition $t \in \mathcal{T}^T$ is drawn as a rectangle and has an exponentially distributed firing time. An (immediate) transition $i \in \mathcal{T}^I$ is drawn as a thin bar and has a constant zero firing time.
- $a : ((\mathcal{P}^D \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P}^D)) \times \hat{\mathcal{S}} \rightarrow \mathcal{N}$ and $a : ((\mathcal{P}^C \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P}^C)) \times \hat{\mathcal{S}} \rightarrow \mathcal{R}_0$ describe the marking-dependent cardinality (for discrete places) or the fluid impulse (for continuous places) of the input and output arcs connecting transitions and places. We use the same symbol for both, and we draw them as thin arcs with an arrowhead on their destination, since the type of place eliminates any possibility of confusion. The function describing a is written on the arc, the default is the constant one.

- $f : ((\mathcal{P}^C \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P}^C)) \times \hat{\mathcal{S}} \rightarrow \mathcal{R}_0$ describes the marking-dependent fluid rate of the input and output arcs connecting transitions and continuous places. These fluid arcs are drawn with a thick line, and an arrowhead on their destination. Also in this case the function is written on the arc and the default is the constant one.
- $g : \mathcal{T} \times \hat{\mathcal{S}} \rightarrow \{0, 1\}$ describes the marking-dependent guard of each transition.
- $\lambda : \mathcal{T}^T \times \hat{\mathcal{S}} \rightarrow \mathcal{R}_0$ and $w : \mathcal{T}^I \times \hat{\mathcal{S}} \rightarrow \mathcal{R}_0$ describe the marking-dependent firing rates and weights of each transition.
- $b : \mathcal{P}^C \times \mathcal{N}^{|\mathcal{P}^D|} \rightarrow \mathcal{R}_0 \cup \{\infty\}$ describe the fluid bounds on each continuous place. This bound has no effect when it is set to infinity. Note that b depends only on the discrete part of the state space, $\mathcal{N}^{|\mathcal{P}^D|}$, not on $\hat{\mathcal{S}}$, to avoid the possibility of circular definitions.
- $(\mathbf{m}^0, \mathbf{x}^0) \in \hat{\mathcal{S}}$ is the initial marking. Graphically, it is represented by writing the value of \mathbf{m}_p^0 , or \mathbf{x}_q^0 , inside the corresponding place. A missing value indicates zero. For discrete places, it is also common to draw \mathbf{m}_p^0 tokens inside the place, if this number is small.

The meaning of these quantities is given by the enabling and firing rules. We say that a transition $t \in \mathcal{T}$ has concession in marking (\mathbf{m}, \mathbf{x}) iff

$$\forall p \in \mathcal{P}^D, \quad a_{p,t}(\mathbf{m}, \mathbf{x}) \leq \mathbf{m}_p \quad \text{and} \quad g_t(\mathbf{m}, \mathbf{x}) = 1.$$

If any immediate transition has concession in (\mathbf{m}, \mathbf{x}) , it is said to be enabled and the marking is said to be vanishing. Otherwise, the marking is said to be tangible and any timed transition with concession is enabled in it. In other words, a timed transition is not enabled in a vanishing marking even if it has concession.

Some definitions of SPNs allow one to disable a transition t with concession in a marking by specifying a zero rate or weight for it, or by introducing inhibitor arcs, drawn with a circle instead of an arrowhead. Since we can represent these behaviors by an appropriate definition of input arc cardinalities or of guards, we assume, without loss of generality, that rates and weights are positive for an enabled transition. Inhibitor arcs can then be considered merely as a shorthand[§].

[§] If, in (\mathbf{m}, \mathbf{x}) , an inhibitor arc from $p \in \mathcal{P}^D$ ($q \in \mathcal{P}^C$) to $t \in \mathcal{T}$ has cardinality $c \in \mathcal{N}$ ($c \in \mathcal{R}^0$), t is disabled if $c \geq \mathbf{m}_p$ ($c \geq \mathbf{x}_q$). The same behavior can be modeled by ensuring that the guard g_t evaluates to 0 in (\mathbf{m}, \mathbf{x}) .

Let $\mathcal{E}(\mathbf{m}, \mathbf{x})$ denote the set of enabled transitions in marking (\mathbf{m}, \mathbf{x}) . Enabled transitions change the marking in two ways. First, a transition $t \in \mathcal{T}$ enabled in marking (\mathbf{m}, \mathbf{x}) can fire after a random amount of time having distribution $\sim \text{Expo}(\lambda_t(\mathbf{m}, \mathbf{x}))$, and yield a (possibly) new marking $(\mathbf{m}', \mathbf{x}')$. We then write $(\mathbf{m}, \mathbf{x}) \xrightarrow{t} (\mathbf{m}', \mathbf{x}')$, where

$$\begin{aligned} \forall p \in \mathcal{P}^D, \quad \mathbf{m}'_p &= \mathbf{m}_p + a_{t,p}(\mathbf{m}, \mathbf{x}) - a_{p,t}(\mathbf{m}, \mathbf{x}) \\ \forall q \in \mathcal{P}^C, \quad \mathbf{x}'_q &= \min\{b_q(\mathbf{m}'), \max\{0, \mathbf{x}_q + a_{t,q}(\mathbf{m}, \mathbf{x}) - a_{q,t}(\mathbf{m}, \mathbf{x})\}\}. \end{aligned}$$

Second, fluid flows continuously through the arcs f of enabled transitions connected to continuous places. The potential rate of change of fluid level for the continuous place $q \in \mathcal{P}^C$ in marking (\mathbf{m}, \mathbf{x}) is

$$\delta_q^{pot}(\mathbf{m}, \mathbf{x}) = \sum_{t \in \mathcal{E}(\mathbf{m}, \mathbf{x})} f_{t,q}(\mathbf{m}, \mathbf{x}) - f_{q,t}(\mathbf{m}, \mathbf{x}).$$

However, the fluid level can never become negative or exceed the bound $b_q(\mathbf{m})$, so the (actual) rate of change over time, τ , while in marking (\mathbf{m}, \mathbf{x}) , is

$$\delta_q(\mathbf{m}, \mathbf{x}) = \frac{d\mathbf{x}_q}{d\tau} = \begin{cases} 0 & \text{if } b_q(\mathbf{m}) = 0 \text{ or} \\ & \text{if } \mathbf{x}_q = b_q(\mathbf{m}) \text{ and } \delta_q^{pot}(\mathbf{m}, \mathbf{x}) > 0 \text{ or} \\ & \text{if } \mathbf{x}_q = 0 \text{ and } \delta_q^{pot}(\mathbf{m}, \mathbf{x}) < 0 \\ \delta_q^{pot}(\mathbf{m}, \mathbf{x}) & \text{otherwise} \end{cases}. \quad (1)$$

The stochastic evolution of the FSPN in a tangible marking is governed by a race [2]: the timed transition t with the shortest firing time is the one chosen to fire next, unless it becomes disabled by some fluid levels reaching particular values that cause t to become disabled prior to its firing. In a vanishing marking, instead, the weights are used to decide which transition should fire: an immediate transition u enabled in marking (\mathbf{m}, \mathbf{x}) fires with probability

$$\frac{w_u(\mathbf{m}, \mathbf{x})}{\sum_{u' \in \mathcal{E}(\mathbf{m}, \mathbf{x})} w_{u'}(\mathbf{m}, \mathbf{x})}. \quad (2)$$

3 General case

The FSPN definition we just gave is very powerful, but it allows one to describe models whose solution can be quite difficult, even with discrete-event simulation. Indeed, it can be used to define FSPNs whose behavior is “unstable,” as in the FSPNs of Fig. 1. In the model

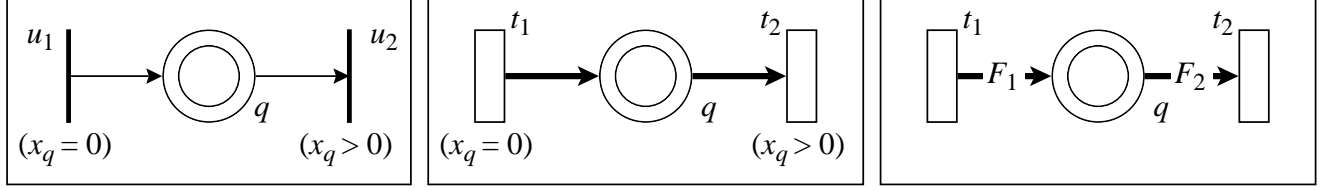


Figure 1: FSPNs exhibiting unstable behaviors.

on the left, immediate transitions u_1 and u_2 alternatively put and remove a unit impulse instantaneously. With few exceptions [8], such a behavior has been considered a modeling error in the literature on discrete-state models. The instability of the model in the middle is instead exclusive to models with a states having a continuous component, such as our FSPNs. When $\mathbf{x}_q^0 = 0$, timed transition t_1 is enabled and timed transition t_2 is disabled. However, as soon as the fluid arc starts adding fluid to q , the situation is reversed, t_1 becomes disabled, while t_2 becomes enabled and starts emptying q . It could be argued that, in such a situation, q will always be empty, but any model where an infinite number of events occurs in a finite time (e.g., transitions t_1 and t_2 become enabled an infinite number of times) cannot be managed by conventional discrete-event simulation techniques. Hence, we will consider such behaviors illegal.

The model on the right could be also considered unstable if $F_2 > F_1$. Both t_1 and t_2 are always enabled, hence there is a continuous flow into q at rate F_1 due to t_1 . However, the outgoing flow due to t_2 cannot be F_2 . Our definition simply states that δ_q is 0 in this case, implying that the outgoing flow is effectively reduced to be F_1 , instead of F_2 , or, in other words, the arc from q to t_2 can be thought to have effect only a fraction F_1/F_2 of the time. This type of behavior, however, can be easily managed by considering all the flows incident to a continuous place, so we do not regard it as a true instability.

We now describe how to study a model with no unstable behaviors. Assume that we have just entered tangible marking (\mathbf{m}, \mathbf{x}) . If there is any enabled transition, each continuous component \mathbf{x}_q might vary in a very general way over time. Applying Eq. 1 to each $q \in \mathcal{P}^C$, we obtain a system of ordinary differential equations subject to the initial condition $\mathbf{x}(0) = \mathbf{x}$. We can then consider two cases:

- In the simpler case, the cardinality of the arcs connected to discrete places and the guards do not depend on \mathbf{x} . Even so, the firing times behave as a nonhomogeneous

Poisson process (NHPP) whose rate depends on the continuous marking, and so some care is required in sampling the firing instants. We assume that the firing rate of each transition t can be bounded from above by $\lambda_t^*(\mathbf{m})$, given our knowledge of its dependence on the fluid marking. That is, when the discrete marking is \mathbf{m} , the rate of t satisfies $\lambda_t(\mathbf{m}, \mathbf{x}) \leq \lambda_t^*(\mathbf{m})$, for any value of \mathbf{x} that might be reached in conjunction with \mathbf{m} . We can therefore sample from the NHPP using the technique of “thinning” [10], where we sample “potential firing instants” in accordance with a homogeneous Poisson arrival process with rate

$$\Lambda^*(\mathbf{m}) = \sum_{t \in \mathcal{E}(\mathbf{m}, \mathbf{x})} \lambda_t^*(\mathbf{m}).$$

From this process, we can define a sequence of increasing time instants (τ_1, τ_2, \dots) . Starting from $i = 1$, we “accept” τ_i , that is, we declare that a firing occurred at time τ_i , with probability $\Lambda(\mathbf{m}, \mathbf{x}(\tau_i)) / \Lambda^*(\mathbf{m})$, where

$$\Lambda(\mathbf{m}, \mathbf{x}(\tau_i)) = \sum_{t \in \mathcal{E}(\mathbf{m}, \mathbf{x})} \lambda_t(\mathbf{m}, \mathbf{x}(\tau_i)).$$

In other words, we use the actual firing rates at time τ_i as a weight, to determine whether the event corresponds to a true firing or not. This requires us to solve for the value of $\mathbf{x}(\tau_1)$, by integrating the system of ordinary differential equations. If τ_1 is accepted, we stop. Otherwise, we integrate until τ_2 , compute $\mathbf{x}(\tau_2)$, and decide whether to accept τ_2 or not, and so on. Eventually, this process stops at some step i , giving us the actual firing time $\tau^f = \tau_i$.

For example, Fig. 2 illustrates the case where four transitions are enabled in (\mathbf{m}, \mathbf{x}) , t_1 , t_2 , t_3 , and t_4 . The sequence of numbered arrows shows the random deviates generated, in order. First, we generate τ_1 according to the distribution $\text{Expo}(\Lambda^*(\mathbf{m}))$. Then we generate a random deviate $\sim \text{Unif}(0, \Lambda^*(\mathbf{m}))$. In the figure, this happens to fall in the interval corresponding to the “do not accept” case. Thus, we need to generate another potential firing time by sampling the distribution $\text{Expo}(\Lambda^*(\mathbf{m}))$ again and summing the sampled value to τ_1 , obtaining τ_2 . We also need another random deviate $\sim \text{Unif}(0, \Lambda^*(\mathbf{m}))$, which also, in the figure, happens to cause a rejection. Finally, we generate a third potential firing time, τ_3 . When we sample $\sim \text{Unif}(0, \Lambda^*(\mathbf{m}))$, we obtain a value falling in the interval corresponding to t_2 , hence we schedule the firing of t_2 at time τ_3 . It is then apparent that the expected number of random deviates that need to be generated is larger when the bounds $\lambda_t(\mathbf{m}, \mathbf{x})$ for the enabled transitions

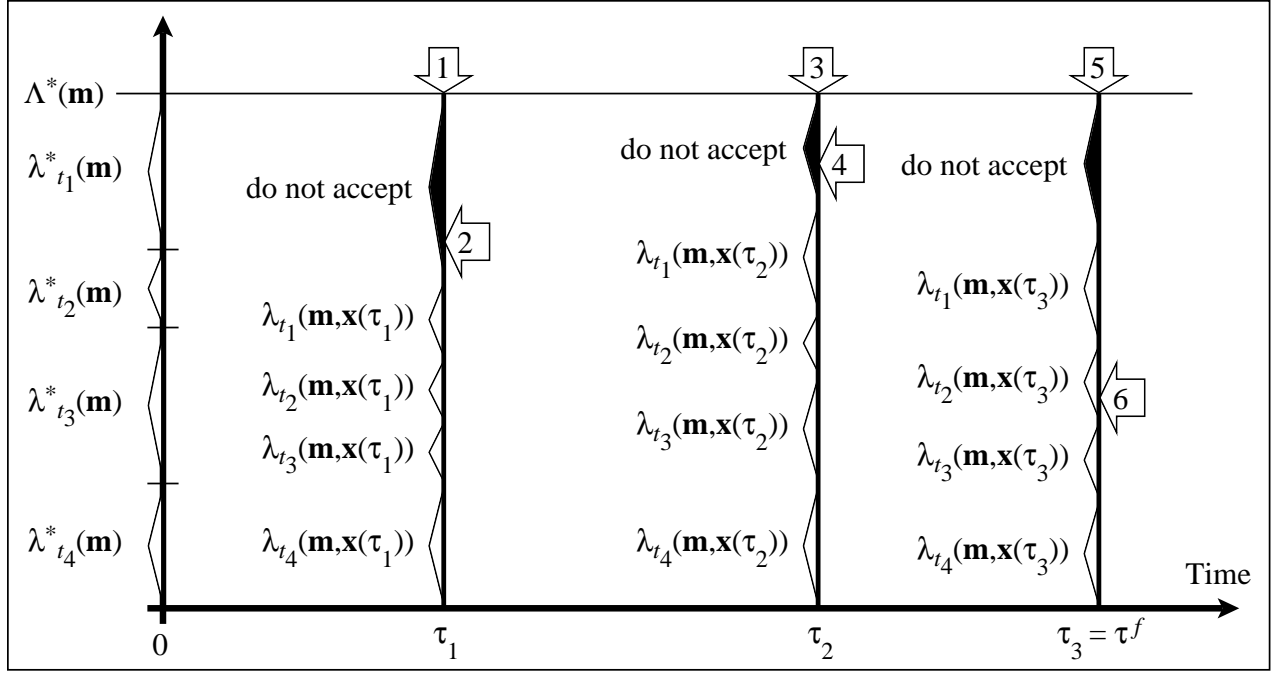


Figure 2: Sampling the NHPP process underlying a FSPN.

are less tight. On the other hand, if the rates of the enabled transitions are a function of \mathbf{x} , but $\sum_{t \in \mathcal{E}(\mathbf{m}, \mathbf{x})} \lambda_t(\mathbf{m}, \mathbf{x})$ is a known constant independent of \mathbf{x} , only two deviates are needed: the first one to decide τ_1 and the second one to decide which transition to fire.

- If, instead, the set of enabled transitions can change as \mathbf{x} evolves, we also need to consider an “enabling event” at the time τ^e when the first change in $\mathcal{E}(\mathbf{m}, \mathbf{x})$ occurs. The method to compute τ^e depends on the nature of the dependencies. In principle, we should know the value of $\mathbf{x}(\tau)$ over the entire horizon $\tau \in [0, \tau^f]$. This can still be accomplished through integration. After (during) integration we need to find that value of τ^e that first satisfies the given condition on the fluid levels. If there is no minimum value $\tau^e \in [0, \tau^f]$ for which the set of enabled transitions changes, the next event to schedule is the firing at time τ^f . Otherwise, we must schedule an “enabling event” at time τ^e .

In either case, if the firing rates of timed transitions are not dependent on fluid levels, the generation of next firing times is considerably simplified because the machinery of NHPP-

based generation of random deviates is avoided.

The processing of the scheduled event causes a change of marking, from (\mathbf{m}, \mathbf{x}) to $(\mathbf{m}', \mathbf{x}')$, where $\mathbf{m}' = \mathbf{m}$ if the event was of the enabling type. Then, in marking $(\mathbf{m}', \mathbf{x}')$, a finite sequence of immediate firings might take place, just as in ordinary, non-fluid, SPNs, until the next tangible marking $(\mathbf{m}'', \mathbf{x}'')$ is reached. Thanks to the memoryless property of the exponential distribution, the evolution of the FSPN from this point on is analogous to the evolution from the initial marking, that is, we do not need to be concerned about the “remaining firing times” of transitions that were already enabled prior to reaching this marking.

4 Uncoupled behavior

The general behavior just described requires us to solve a system of ordinary differential equations at each step of the simulation. This computation can be quite costly. A restriction on the type of dependency allows us to uncouple the system, resulting in a set of ordinary differential equations which can be solved independently. This requires that the fluid rates incident on q , hence $\delta_q(\mathbf{m}, \mathbf{x})$, depend only on $(\mathbf{m}, \mathbf{x}_q)$, not on the fluid levels in the other continuous places:

$$\forall (\mathbf{m}, \mathbf{x}), (\mathbf{m}, \mathbf{x}') \in \hat{\mathcal{S}}, \quad \mathbf{x}_q = \mathbf{x}'_q \Rightarrow \delta_q(\mathbf{m}, \mathbf{x}) = \delta_q(\mathbf{m}, \mathbf{x}').$$

As in the general case, we can still distinguish whether the set of enabled transitions can be affected by \mathbf{x} or not, and the NHPP random variate generation must be used only if their firing rates depend on \mathbf{x} .

5 Predefined classes of behaviors

For particular cases of uncoupled dependencies, we can even have a built-in closed form solution, which will avoid the need for numerical integration altogether. One such case is when, in a given marking (\mathbf{m}, \mathbf{x}) ,

$$\frac{d\mathbf{x}_q(\tau)}{d\tau} = A(\mathbf{m})\mathbf{x}_q(\tau) + B(\mathbf{m}), \quad A(\mathbf{m}) \neq 0$$

whose solution is

$$\mathbf{x}_q(\tau) = -\frac{B(\mathbf{m})}{A(\mathbf{m})} + \left(\mathbf{x}_q(0) + \frac{B(\mathbf{m})}{A(\mathbf{m})} \right) e^{A(\mathbf{m})\tau},$$

assuming that \mathbf{x}_q remains between 0 and $b_q(\mathbf{m})$ during $[0, \tau]$. This answers the question of how much the fluid level in a place will change during the firing time τ of a timed transition. Inversely, the time τ_q when place q reaches a certain fluid level threshold L_q is given by

$$\tau_q = \frac{\ln \left(\frac{L_q + \frac{B(\mathbf{m})}{A(\mathbf{m})}}{\mathbf{x}_q(0) + \frac{B(\mathbf{m})}{A(\mathbf{m})}} \right)}{A(\mathbf{m})},$$

if this quantity is positive (if it is negative, we can simply define $\tau_q = \infty$, that is, the threshold L_q cannot be reached in this marking).

If the set of enabled transitions can only change when some place q reaches a threshold level L_q , then we can simply define the time τ^e of the next enabling event as

$$\tau^e = \min_{q \in \mathcal{P}^C} \{\tau_q\}.$$

When $A(\mathbf{m}) = 0$, the solution is much simpler,

$$\frac{d\mathbf{x}_q(\tau)}{d\tau} = B(\mathbf{m}) \quad \Rightarrow \quad \mathbf{x}_q(\tau) = \mathbf{x}_q(0) + B(\mathbf{m})\tau,$$

again assuming that \mathbf{x}_q remains between 0 and $b_q(\mathbf{m})$ during $[0, \tau]$. The time τ_q when place q reaches the threshold L_q is then

$$\tau_q = \frac{L_q - \mathbf{x}_q(0)}{B(\mathbf{m})},$$

if this quantity is positive, infinity otherwise.

6 Piecewise constant behavior

Complete dependency on the marking (\mathbf{m}, \mathbf{x}) is desirable in principle, but the complication it entails is often excessive and its full power unneeded. A simpler type of dependency is obtained by enforcing a discretization on the behavior of the FSPN with respect to the continuous component \mathbf{x} . This can be accomplished by defining a set of boolean threshold-type conditions $\mathcal{L} = \{(r_1 \odot_1 l_1), \dots, (r_{|\mathcal{L}|} \odot_{|\mathcal{L}|} l_{|\mathcal{L}|})\}$, where $r_k \in \mathcal{P}^C$ is a continuous place, $\odot_k \in \{<, \leq, =, \geq, >, \neq\}$ is a comparison operator, and $l_k : \mathcal{N}^{|\mathcal{P}^D|} \rightarrow \mathcal{R} \cup \{\infty\}$ is a threshold dependent on the discrete marking only. Hence, given a marking (\mathbf{m}, \mathbf{x}) , we can define

the “discretized” marking (\mathbf{m}, \mathbf{i}) obtained through \mathcal{L} , where $\mathbf{i} \in \{0, 1\}^{|\mathcal{L}|}$, and $\mathbf{i}_k = 1$ iff $\mathbf{x}_{r_k} \odot_k l_k(\mathbf{m})$.

If we force a (for discrete places only), f , g , and λ to be defined on the discretized marking (\mathbf{m}, \mathbf{i}) , rather than on (\mathbf{m}, \mathbf{x}) , then the behavior of the FSPN is constant until the first threshold is encountered, or until a firing occurs.

Hence, we can carry on a traditional discrete-event simulation, where the types of events that need to be scheduled in the event queue are either transition firings or the hitting of a threshold.

Fortunately, there is no need to place the same restriction on the fluid impulses (a for continuous places) or w , since the impulses and the weights are always evaluated only at a specific instant in time. Applying the restriction to these quantities as well would prevent us from modeling useful behaviors, such as emptying a continuous place, or choosing between two immediate transitions with probability proportional to the level in two continuous places.

7 Examples

We illustrate the power of the formalism with a few examples.

7.1 A queue with impatient customers and breakdowns

Consider a queue with a server subject to breakdowns and repairs. The customers arrive with a constant rate, and queue in an unbounded waiting room. They are served in first-come-first-serve order, but, once their service starts, they can become impatient and leave before completion, see Fig. 3. Unlike other system with impatient customers, the amount of time a customer has been in the queue before his service begins does not affect his decision to leave. The arcs from *Serving* to *Busy* and from *Waiting* to *Idle* are used to count time into the two places, hence they have fluid rate one. The arcs from *Busy* and *Idle* to *Serving* have impulse \mathbf{x}_{Busy} and \mathbf{x}_{Idle} defined on them, respectively. Hence, they are “flushing” arcs, they have the effect of emptying the two places immediately after the firing of *Serving*.

The guard of immediate transition *Leave* specifies when the customer at the head of the queue decides to leave. Various policies can be easily modeled:

- The total amount of time from the moment service began exceeds a certain threshold MAX . Then, we could define the guard g_{Leave} to be the boolean expression $(\mathbf{x}_{Busy} + \mathbf{x}_{Idle} = MAX)$.

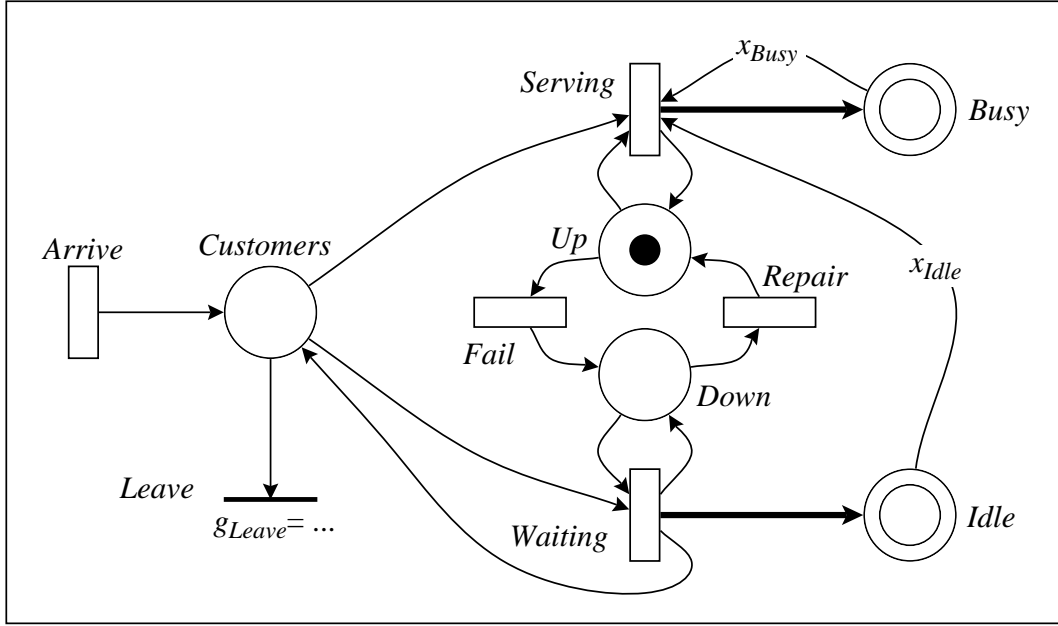


Figure 3: The FSPN of a queue with impatient customers and breakdowns.

This policy is representative of situations where, once the server begins operating on a customer, the operation must complete within a certain time, to avoid spoilage, etc.

- The total amount of time a customer has not received any service from the moment service began exceeds a certain threshold MAX . Then, $g_{Leave} = (\mathbf{x}_{Idle} = MAX)$.

This could represent a similar situation, where, spoilage occurs only when the customer is not being served.

- A customer has waited for an uninterrupted period of time MAX without receiving any service. Then, $g_{Leave} = (\mathbf{x}_{Idle} = MAX)$, after adding an impulse arc $a_{Busy, Repair}(\mathbf{m}, \mathbf{x}) = \mathbf{x}_{Busy}$, so that place *Busy* becomes empty after each repair.

This could represent a situation, where, in addition to occurring only when the customer is not being served, any spoilage immediately disappears as soon as service resumes.

- A customer has spent more time waiting for the server to be operational than receiving service, from the moment service began. Then, $g_{Leave} = (\mathbf{x}_{Idle} > \mathbf{x}_{Busy})$.

A measure of interest is the fraction of jobs that decide to leave:

$$\frac{\text{number of firings of } \textit{Leave} \text{ up to time } \tau}{\text{number of firings of } \textit{Arrive} \text{ up to time } \tau}$$

computed over a finite horizon, or in the limit for $\tau \rightarrow \infty$.

7.2 A dual-tank processing facility

Consider a processing plant where, during normal operation, a liquid enters a main tank, *One*, from an external source with rate γ_{in} , and is used by a processing station, with a (potential) rate $\gamma_{out} > \gamma_{in}$, (see Fig. 4).

However, the processing station is subject to breakdowns during which it cannot process the liquid. Interrupting the flow from the external source of liquid into the main tank is an expensive operation, hence, a second additional tank, *Two*, is present. When the processing station is down, the liquid is automatically routed to tank *Two*, which has a maximum capacity b_{Two} . Only when the second tank is full, the flow from the external source is shut down. After a repair, the processing can resume and the liquid is routed again from the external source, which is restarted if it had been shut down, into tank *One*. In addition, any liquid in tank *Two* is pumped into tank *One*, with rate γ_{12} . If $\gamma_{in} + \gamma_{12} > \gamma_{out}$, the level in tank *One* will increase while the processing station is working to catch up after a repair. Since tank *One* has a maximum capacity, b_{One} , the flow from tank *Two* to tank *One*, rather than the flow from the external source, is slowed down when this limit is reached. The guard (in the FSPN of Fig. 5) $g_{Xfer} = (\mathbf{x}_{One} < b_{One})$ accomplishes this.

The main reason for having two tanks, instead of a single large one, is efficiency. As the liquid needs to be maintained at a given temperature, tank *One* is constantly heated, while tank *Two* is heated only when it contains liquid, i.e., during a breakdown. Indeed, the two measures we are interested in computing are:

$$\frac{\text{number of firing of } \textit{Start} \text{ up to time } \tau}{\tau},$$

the frequency at which the external source needs to go through a start-stop cycle, and

$$\text{probability that tank } \textit{Two} \text{ is not empty at time } \tau,$$

again, either for a finite τ or in the limit for $\tau \rightarrow \infty$.

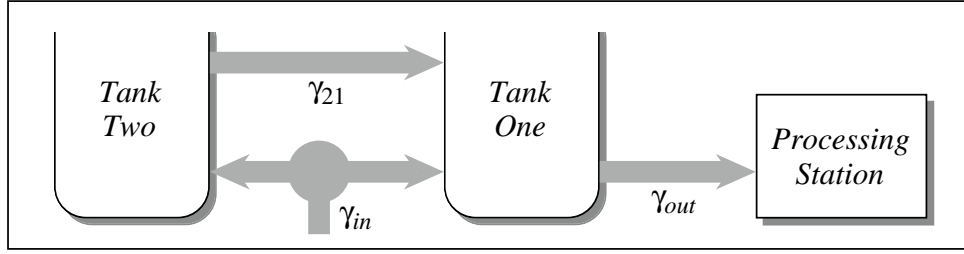


Figure 4: A dual-tank processing facility.

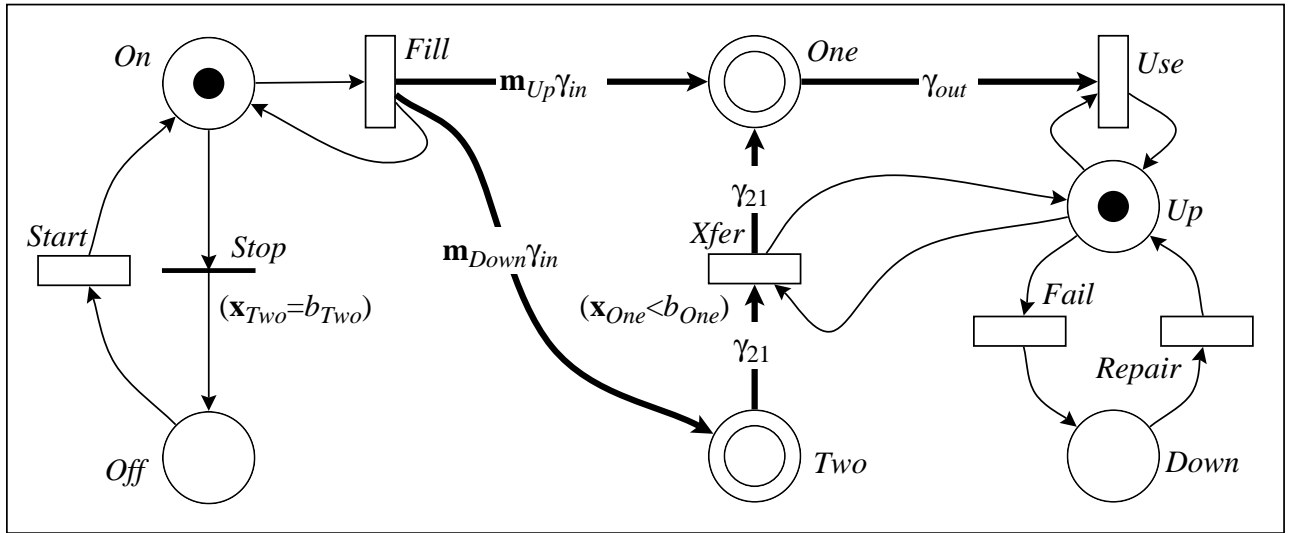


Figure 5: The FSPN of the dual-tank processing facility.

8 Conclusion

In this paper we have extended the power of recently introduced fluid stochastic Petri nets. Since equations characterizing the evolution of such FSPNs are a coupled system of partial differential equations, the generation and solution of these equations can become intractable except for small or very well structured FSPNs. Hence, discrete-event simulation becomes an important avenue for the solution of FSPNs. Because of mixed, discrete and continuous state space with heavy interactions between them, simulation also poses some challenges.

Some of the challenges are addressed in the paper. Actual implementation (currently in progress) of an FSPN simulator will undoubtedly reveal other problems.

References

- [1] M. Ajmone Marsan, G. Balbo, and G. Conte, *Performance Models of Multiprocessor Systems*, MIT Press, Cambridge, MA, 1986.
- [2] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani, “The effect of execution policies on the semantics and analysis of Stochastic Petri Nets,” *IEEE Trans. Softw. Eng.*, vol. 15, pp. 832–846, July 1989.
- [3] D. Anick, D. Mitra and M. Sondhi, “Stochastic Theory of Data-Handling Systems,” *The Bell System Technical Journal*, Vol. 61, No. 8, pp. 1871-1894, Oct. 1982.
- [4] C. G. Cassandras, *Discrete Event Systems: Modeling and Performance Analysis*, Aksen Associates, Holmwood, IL, 1993.
- [5] G. Ciardo, “Discrete-time Markovian stochastic Petri nets,” in *Numerical Solution of Markov Chains '95* (W. J. Stewart, ed.), (Raleigh, NC), pp. 339–358, Jan. 1995.
- [6] G. Ciardo, J. K. Muppala, and K. S. Trivedi, “Analyzing concurrent and fault-tolerant software using stochastic Petri nets,” *J. Par. and Distr. Comp.*, 15(3):255–269, July 1992.
- [7] A. I. Elwalid and D. Mitra, “Statistical Multiplexing with Loss Priorities in Rate-Based Congestion Control of High-Speed Networks,” *IEEE Transaction on Communications*, Vol. 42, No. 11, pp. 2989-3002, November 1994.
- [8] W. K. Grassmann and Y. Wang, “Immediate events in Markov chains,” in *Numerical Solution of Markov Chains '95* (W. J. Stewart, ed.), (Raleigh, NC), pp. 163–176, Jan. 1995.
- [9] G. Horton, V. Kulkarni, D. Nicol, and K. Trivedi, “Fluid stochastic Petri nets: Theory, application, and solution,” ICASE Report 96-5, Institute for Computer Applications in Science and Engineering, Hampton, VA, 1996.

- [10] P.A.W. Lewis and G.S. Shedler, “Simulation of Nonhomogeneous Poisson Processes by Thinning,” *Naval Research Logistics Quarterly*, Vol. 26, pp. 403-414, 1979.
- [11] D. Mitra, “Stochastic Theory of Fluid Models of Multiple Failure-Susceptible Producers and Consumers Coupled by a Buffer,” *Advances in Applied Probability*, Vol. 20, pp. 646-676, 1988.
- [12] T. Robertazzi, *Computer Networks and Systems: Queueing Theory and Performance Evaluation*, Springer-Verlag, 1990.
- [13] M. Telek, A. Bobbio, and A. Puliafito, “Steady state solution of MRSPN with mixed preemption policies,” in *Proc. IEEE International Computer Performance and Dependability Symposium (IPDS’96)*, (Urbana-Champaign, IL, USA), pp. 106–115, IEEE Comp. Soc. Press, Sept. 1996.
- [14] K. S. Trivedi and V. G. Kulkarni, “FSPNs: fluid stochastic Petri nets,” in *Proc. 14th Int. Conf. on Applications and Theory of Petri Nets*, (Chicago, IL), pp. 24–31, June 1993.
- [15] N. Viswanadham and Y. Narahari, *Performance Modeling of Automated Manufacturing Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1992.